CIRCULO USUARIOS QL NUMERO 2 15 NOVIEMBRE 1988.

INDICE
======

EDITORIAL

Otra vez estoy a pie del cañòn, que remedio. Y voy a contaros una pequeña historieta.

Hacen falta colaboraciones. Para llenar este boletín he agotado todo lo que he podido conseguir, no hay nada más. Pero si los socios envian material, yo podria confeccionar otro boletín antes de fin de año, pero no puedo dar ninguna garantia sobre lo que tiene pensado hacer Diego Alcalá.

El último boletín estaba casi totalmente dedicado al Forth y al ratón de Inves. Pero os puedo asegurar que era todo el material que tenia a mano (el boletín estaba confeccionado totalmente por dos socios de FIG).

El futuro del QL depende actualmente totalmente de los usuarios QL. Hay que arrimar el codo y cooperar en escribir software para nuestra máquina, pues no será porque no tenemos compiladores, ensambladores, lenguajes, toolkit,.... Lo que si es cierto es que la mayoria de nosotros no sabemos o creemos que no sabemos programar, pero el verdadero objetivo de QLave siempre ha sido que los usuarios aprendan de otros usuarios (el intercambio de información e ideas).

SI TODOS LOS SOCIOS HUBIESEN ESCRITO SOLAMENTE DOS PAGINAS POR AÑO, HABRIA COLABORACIONES DE SOBRA PARA PUBLICAR TODOS LOS NUMEROS DE QLAVE MENSUALES DE UN AÑO.

SI TODOS LOS SOCIOS HUBIESEN CEDIDO UN PEQUEÑO PROGRAMA POR AÑO, EN LA LIBRERIA SE HABRIA MULTIPLICADO EL NUMERO DE TITULOS POR TRES.

SI DIEGO FUESE MAS INTELIGENTE Y RESPONSABLE, NO ESTARIAMOS EN EL ESTADO ACTUAL.

Si hoy en España hiciesemos una encuesta preguntando a los usuarios de ordenadores (a otra gente mejor no preguntar) : ¿ Qué es un QL ? . De cada 100, uno diria que escuchó una vez hace mucho tiempo algo, los otros 99 ni siquiera saben que existió.
Si a esos mismos usuarios o posibles usuarios preguntasemos ¿ qué ordenador para trabajar profesionalmente comprarian ?. De cada 100, 80 comprarian un PC, y los otros 20 un 68K (se repartiria entre MAC, Atari ST y Amiga).

Como podreis observar, todos los datos son ficticios, pero son casi exactos con la realidad actual.

Los usuarios de QL ampliado en RAM y unidad de disco estamos muy contentos con nuestra configuración y podemos esperar a que nuestro QL explote o aparezcan en el mercado los ordenadores del futuro (arquitecturas como la RISC, transputers,...) o un sucesor QL basado en el MC 68030.

Las colaboraciones deben ser enviadas a:

Salvador Merino
Ctra. Cádiz, Cerámicas Mary
29640 Torreblanca del Sol
Fuengirola (Málaga)
Teléfono 475043

Prometo, aunque sé que me va a costar dinero de mi bolsillo, devolver los discos un dia o dos después de su recepción (por favor, cuando reciban los discos de retorno, no borren sus ficheros hasta que vean publicados sus articulos).

¿ Cuál es el formato de las colaboraciones ?

Son ficheros QUILL con el margen izquierdo a 0, el sangrado a 0, y el derecho a 80. Máximo 45 lineas por página. No pie de página. Solamente usar caracteres normales (No negrita, subrayado,.....).

¿ Qué se puede enviar ?.

Muy simple cualquier cosa (Cartas, preguntas, programas, trucos, comentarios, criticas, anuncios, articulos.... en otras palabras, DE TODO, y sin censura).

Junto a este fichero no he podido enviar ningún nuevo programa para la libreria, pero el boletín está plagado de varias rutinas en varios lenguajes.

Si QLave desaparece, hay proyectos de nuevos Clubs (actualmente hay tres proyectos muy diferentes).

S. Merino

CARTAS DE LOS SOCIOS

NOTA:

La carta anterior fue escrita hace ya muchos meses, aunque por diversos problemas no se ha podido publicar hasta ahora. Eso explica que tantas cosas hayan quedado defasadas. En algunos casos los acontecimientos han ido por el camino que yo hubiera deseado, en otros queda mucho por hacer, en otros aún las cosas han seguido por caminos totalmente impensados, unas veces mejores de los que yo señalaba y otras no tanto.

José Carlos de Prada

ANUNCIO :

Si alguien está interesado en formar parte de un Club de colaboradores que confeccionan un boletín QUILL de 100 K mensual y escriben programas, pida más detalles a Salvador Merino (Teléfono (952) 475043). Aún quedan 15 plazas libres, y lo más interesante es que no hay que pagar cuota debido a que no existen gastos.

S. Merino

PREGUNTAS

Una vez leido un fichero con el comando RU en The Editor, ¿cómo insertar en él caracteres no directamente accesibles por el teclado? Por ejemplo, aquellos que se corresponden con acciones a realizar por The Editor, como los caracteres ASCII correspondientes a cada cursor. ¿Alguien sabe cómo hacer esto? Por más que releo el manual, no encuentro nada al respecto.

Más problemas con la TRUMP CARD. Como sabréis, la Trump Card incluye un buffer dinámico de impresora, aunque se puede emplear para varias cosas. Funciona interceptando las aperturas de ficheros que se dirijan a cierto dispositivo seleccionado y dirigiendo el flujo de datos a una zona propia para gestionarlo por su cuenta. Sin hacer nada de nada, al encender el ordenador, tenemos un buffer para las salidas serie si abrimos los ficheros al dispositivo "prt". Por ejemplo, si hacemos "OPEN#3,prt1" es como si hiciésemos "OPEN#3,ser1" pero con un buffer, de modo que si hacemos después "LIST#3" podemos olvidarnos de esperar a que el listado termine, pues queda en el buffer y podremos dedicarnos a otras cosas.
Mediante el comando PRT_USE de la Trump Card podemos alterar el dispositivo al que se destina el buffer (ser), así como el nombre identificativo de este último (prt). Por ejemplo, al hacer "PRT_USE ser,ser" creamos un buffer transparente para ambas salidas serie, pues cualquier salida hacia ellas será interceptada y gestionada por el buffer.
Hasta aquí todo perfecto. Pero hay algo que no funciona. Si hemos abierto un fichero a ser1 o ser2 con buffer, ¡¡¡no se puede recibir por él!!! Tanto INPUT# como INKEY# o cualquier otra cosa provocan el error "parámetro incorrecto" o algo parecido, tanto en SuperBasic interpretado como compilado con Turbo.
Para colmo, el comando PRT_USE es incompilable con Turbo, pues al ejecutarse la tarea provoca un error similar.
He descubierto esto como consecuencia de tener conectados el QL y un Spectrum con el objeto de transmitir el código objeto resultante del Ensamblador Z80 de aquél a éste. En un principio transmitía el código tal cual, pero de vez en cuando no resultaba muy fiable. Así, diseñé un protocolo de comunicación para verificar y corregir los posibles errores y entonces aparecieron los problemas, pues el QL no puede recibir nada si el buffer está activado... ¡y si lo desactivo la ejecución del ensamblador se vuelve interminable! ¿Alguien sabe algo de esto?

Marcos Cruz (QLave-242)
Madrid, Julio 1988

PAMPLONA, 4 NOV 88

Hola, Salvador :

Te escribo para hacerte una seria consulta sobre mi "Futuro Desarrollo Informático".
Resulta que varios compañeros de arquitectura, a los que yo convencí para que se compraran un QL; han decidido dar un giro de 180 grados y comprarse un PC Philips.
El asunto está en que los programas para arquitectura en PC son superabundantes y superbuenos. Y como se comparten, el precio es bajo.
Yo les recriminé su actitud, naturalmente, y les recordé la multitarea, el QDOS.... y que además aparecerían unos emuladores de MS-DOS (se me rien a la cara).
Empecé a sudar frio cuando recibí tu carta diciendo que lo de los emuladores no era en modo gráfico y no tenian una fiabilidad grande. Luego ponias lo de que se podia acoplar a un Spectrum compatible PC.
Yo estoy comprandome un monitor. he mirado un Philips color y no acaba de acoplarse la imagen.
Probados un Philips F.V., Hantarex F.V. y BMC F.V.. Este es el mejor con diferencia. He estado reticente a comprarlo por la mala fama que tiene, y me he enterado por algunos socios que lo tienen; que el F.V. es muy bueno, y el de color, un timo.
El siguiente paso seria comprar la Trump Card + U.D. 3.5" (¿ Qué marca ? ¿ DS/DD ?). Esto saldria por 64.000 ptas. Además tu aconsejas la Q-power ¿ cuanto vale ?.
Y digo yo :

¿ El Thor XVI seria compatible con el PC ?
¿ Qué ventajas tendria comprarme el Spectrum compatible en vez de un PC ?
¿ Si me compro un PC, podria leer con el QL en los discos, o en el PC en los discos del QL ?
¿ Serviria el mismo monitor ?
¿ La misma impresora ?
¿ Qué futuro hay a una posible compatiblidad PC-QL ?

Te agradezco tus sabios consejos.

Un saludo:

Fernando Martinez
Pamplona (QLave-143)

ED:

Por lo que me dices, lo único que puedo recomendarte es que te compres un PC barato para poder usar los programas de arquitectura de tus amigos.
Comprar un PC-compatible no es tan fácil como parece. Tienes que tener en cuenta muchas cosas:

- Poder usar con él algunos periféricos que ya poseas (p.e.: monitor, impresora, discos,...).
- Los programas que vas a usar deben correr sin problemas con la tarjeta gráfica que use el PC seleccionado. El IBM PC original no tenia gráficos, y la tarjeta gráfica era opcional.
- También hay que tener en cuenta la versión de MS-DOS y que la compatibilidad con el IBM PC no es total en ningún PC-compatible.
- Cuanto más interfaces posea el PC mejor.
- No sé si se me ha olvidado algo importante, pues nunca he estado interesado en un IBM PC o compatible.

Quizás si no deseas gastar mucho dinero el Sinclair PC 200 con unidad de disco 3.5" DS/DD y 512 K de RAM podria servir de sobra para tus necesidades. El precio de este último está alrededor de 60.000 ptas.

Yo tenia un monitor BMC 1010QL color alta resolución y debido al defecto de fabricación, me he visto obligado a guardarlo en un rincón del almacén y comprar un monitor Philips F.V. que me funciona muy bien y cabe toda la pantalla del QL.
Mi QL ampliado con la Trump Card y una unidad de disco Cumana (NEC) 3.5" DS/DD funciona a las mil maravillas. Tengo un segundo QL (MGE) de repuesto que he usado en red local para probarla (funciona muy bien, es fantastico).
La Q-power es muy cara cuesta 4.000 ptas, pero si tienes problemas de sobrecalentamiento, con cambiar el regulador 7805 por uno de 2 amp que cuesta en comercios especializados sobre 75 ptas, es suficiente.

El THOR XVI solamente podria ser compatible MS-DOS si se escribe un emulador para él o el QL. Aunque el THOR XVI está capacitado para emular la mayoria de las tarjetas gráficas PC.
El Spectrum PC era un rollo propagandistico de Amstrad para llamar la atención del público para su nuevo ordenador Sinclair PC 200 (ya lo hizo anteriormente diciendo que el nuevo Spectrum 2+ tendria los gráficos y sonido del Commodore Amiga).
Si el QL está ampliado con discos puede leer, garantizado con toda seguridad, los discos de un PC de 5.25" y 3.5". Pero el PC no tiene el software preparado para leer los discos en formato QL.
El monitor podria servir si se usa el mismo interface (RGB o video compuesto) y tiene suficiente resolución. Aunque recomiendo probar, antes de comprar.
Si usas una impresora serie en tu QL y el PC tiene salida serie, no veo ningún problema a no ser que los programas PCs estén preparados solamente para impresoras paralelo.

El único futuro que hay para una posible compatibilidad PC-QL es que salga al mercado la Card-FUTURA, y veamos cuales son los PC-compatibles que pueden usarla.

RESPUESTAS

A Fernando Martinez, de Pamplona, QLave-143:
No tengo el ratón de Investrónica, aunque lo estoy esperando de un momento a otro, pero por lo que sé me temo que no es accesible desde Basic sino desde código máquina, y eso siempre que se sepa cómo trabaja a nivel de hardware (lo cual no me fio de que venga en las instrucciones). Desgraciadamente hay varios sistemas diferentes de ratones para el QL, es decir, que no existe un único estándar y por tanto cada ratón de una determinada marca sólo trabajará con los programas que lo tengan previsto, generalmente de esa misma marca, claro. Siento no poder darte más detalles, pero en cuanto tenga el ratón lo comunicaré si indago algo interesante.
Siento igualmente no poder animarte excesivamente ante tus "dudas de fe", y a lo peor para cuando estas lineas aparezcan en QLave ya te has decidido por comprar un ATARI ST antes que ampliar tu QL, pero lo intentaré. Sólo decirte que has de tener en cuenta lo que pretendas sacar del aparato; exceptuándonos a los locos de los bytes, la mayoría de los usuarios pretenden usar sólamente tratamientos de textos, ficheros, hojas de cálculo, algún que otro programa de dibujo y como mucho hacer sus pinitos en Basic, y eso se puede hacer con cualquier ordenador, desde un ZX 81 hasta un Amiga, por poner por caso. El tema surge en el volumen de información que se pretende manejar en cada tarea, y por ello habrá que prever las ampliaciones disponibles para el ordenador que nos hagan más cómodo el trabajo.
Por ejemplo, verás que los anuncios de ventas de las revistas están plagados de ofertas de Spectrum de 48 Kbytes, como hace años lo estaban con ofertas de ZX 81. Cada ordenador tiene una época de esplendor alrededor de su lanzamiento pero su valor no disminuye con el paso del tiempo, sino que simple y lógicamente aparecen otras máquinas de mejores prestaciones y posiblemente a precios más reducidos. Lo curioso es que antes quien vendía un ZX 81 lo hacía para adquirir un Spectrum... ¡Y quien ahora vende el Spectrum es para adquirir un Amstrad CPC!
Un Spectrum con interface de disco, RS232, Centronics, y alguna cosa más como un teclado en condiciones, se convierte en una máquina potente, sin nada que envidiar a bastantes que aparecieron mucho después. Lo mismo le ocurre al QL: ampliado según las necesidades de cada uno no tiene por qué ser destinado a su sutitución por ningún ST ni Amiga. Ten la seguridad de que si tienes la posibilidad de adquirir un 520ST -deduzco que es el 520ST por los 512 Kbytes que citas- a tan bajo precio es porque al fin y al cabo se trata del elemento inferior de una gama, la ST, que a su vez cambiará rápidamente: si compras ahora el 520ST posiblemente dentro de unos meses sentirás algo raro en el estómago al ver ofertas de 1040ST al mismo o menor precio, y quizá con monitor incluido... es la misma vieja historia.
Por lo que dices tu QL es sin expandir y sólo con microdrives. Si has pensado en adquirir una unidad de disco y una ampliación de memoria doy por supuesto que las necesitas, o sea que el volumen de actividad que tienes con el QL es bastante grande. Existen muchas opciones de ampliaciones de memoria, de interfaces y de unidades de disco, y ,aunque me creo que salga más barato un ST con la suya propia, te aconsejo que mires bien en los anuncios del QL World. Con toda seguridad encontrarás ofertas de drives e interfaces más baratas que las de aquí.
Me temo que si a pesar de todo el ST te sale más economico que ampliar el QL no hay nada que hacer... aunque me queda un último recurso:
Debes considerar la cantidad de software del que dispongas, pues un ordenador nuevo supone volver a empezar a comprar programas, y eso es muy importante. Creo que ese es el único aspecto que en último caso podría decidirte en esta alternativa tan dramática. Haz una cuenta de los programas que necesitarías adquirir para trabajar con el Atari ST e infórmate en alguna publicación del precio que vendrían a tener y súmalo al del ordenador...
Como epílogo baste decir que el 520ST es una buena máquina e incluso en algunos aspectos supera al QL. Te listo aquí por curiosidad algunas de sus características:

CPU: 68000 a 8 MHz
MEMORIA: 512 Kbytes de RAM
PANTALLA: tres modalidades, la mayor resolución 640x400 en dos colores
INTERFACES: MIDI (entrada y salida), Centronics, RS-232, para disco, monitor RGB, RF para televisor, para cartucho ROM de 128 Kbytes, dos conectores para joystick o ratón y uno para disco rígido (futuro CD-ROM de 800 Mbytes)

UNIDAD DE DISCO: Sony de 3,5 pulgadas de doble cara (640 Kbytes)
SISTEMAS OPERATIVOS: TOS y GEM
LENGUAJES: ST-BASIC y ST-LOGO

Como verás no se diferencia espectacularmente de un QL ampliado. Uno aventaja al otro en unas cosas y el otro a uno en otras, como siempre.
Bien, espero no haberte dado la lata con todo esto. Al menos, por si no te has decidido ya, mi intención ha sido darte algunos elementos de juicio adicionales desde mi punto de vista particular.

Marcos Cruz (QLave-242)
Madrid, Mayo 1988

EL RATON QUE ERA UN ESCARABAJO PELOTERO

La primera vez que conocí algo sobre el Star Mouse fue en la desaparecida revista Todospectrum, dedicada, como es fácil deducir, al Spectrum, en unas páginas que reservaba de vez en cuando al por entonces novedoso QL. Aquello fue en Diciembre de 1985 y el Star Mouse aparecía como novedad en el SIMO del mismo año. Desarrollado por la empresa española PURICORP y apoyado por Investrónica, se describía al paquete como una utilidad de dibujo muy interesante, si bien no se hacía un estudio serio del mismo ni se comentaban en profundidad sus características más notables. De todo eso ya hace algún tiempo y "todavía" el QL era para mí algo muy cercano... a través de los artículos sobre él en las publicaciones de informática que abundaron en aquella época.
Bien, nos situamos de un golpe a comienzos del presente año. Recibo una carta de una conocida tienda madrileña entre los usuarios del QL. En ella se me hace saber de una oferta entre la que veo la venta del Star Mouse al precio de 1500 pesetas. ¿Será una errata? ¡En el último catálogo de la misma tienda aparecían los Star Mouse a 7500! Llamo por teléfono y, confirmado el precio, les encargo uno contra reembolso. Al fin y al cabo, sabía que no iba a encontrarme con ninguna maravilla, dada la antiguedad del programa, pero si por ese precio obtenía un programa de dibujo con ratón, aunque no fuera excepcional, pues no había mucho que perder... o eso pensaba yo.
Recibo el ratón (en su embalaje original y todo) y me apresuro a sacarlo de la caja. Iba solitariamente acompañado, en el interior de la caja de poliestireno blanco, por un microdrive. El ratón en sí tiene un aspecto algo famélico (tipo MADE IN SPAIN, claro) y hace mirar con sana envidia al que viene con el PC1512, ¡que ya es decir!. El ratón se conecta al ordenador mediante el puerto de ROM y uno de los puertos de joystick, y esto último sí que no lo sospechaba. Inocentemente me preguntaba: ¿será que el ratón simula por hardware las teclas del cursor, correspondientes a uno de los puertos de joystick, y por consiguiente puede emplearse en cualquier programa? Mi sospecha se hizo ilusión cuando comprobé que el único fin por el que había que conectar el ratón al conector de ROM es tomar la alimentación electrica de éste, nada más. No faltaba mucho para que mi sueño se desintegrase en mil pedazos, pero eso lo averigüé más tarde, claro.
La curiosidad me llevó a abrirlo todo: microdrive y ratón, y dos cosas me llamaron la atención: el microdrive que se incluye no tiene ningún tipo de etiquetado relativo al programa, sino simplemente un diminuto papel adhesivo cuadrado en la parte posterior y un número escrito a mano, y por otra parte la bola metálica del ratón parece sacada de una excavación arqueológica de un campamento celta, o mejor se parece a una luna llena que en vez de tener su color habitual estuviese teñida de color óxido y tuviese manchas aisladas de color gris metálico apagado... No sé si estas dos cosas son normales o bien es que yo soy algo ingenuo, pero antes que detenerme a averiguarlo me dirijo raudo al QL, le enchufo el ratón, lo despierto y le hago engullir el microdrive...
¡Oh, sopresa! Al cabo de unos segundos surge un dibujo bastante bien realizado de un enorme ratón que desaparece en un momento y es sustitido por un menú de dos lineas (sobre una pantalla completamente en blanco) para elegir entre las instrucciones o el programa en sí. ¿Es que por aquel entonces los programadores no sabían hacer presentaciones más profesionales? Como poco se podía esperar que el menú se integrase en el dibujo inicial, pero nada de eso: aquél se borra y aparecen en las dos primeras lineas ,"a pelo", las opciones ya mencionadas. Empiezo a ponerme nervioso, ante tamaña chapuza, y elijo la opción de dibujar,

en lugar de leer primero las instrucciones. Este atrevimiento me costó caro, pues había conectado el ratón al puerto de joystick equivocado y aquello no funcionaba. Bueno, tras un par de reseteos y una lectura a las instrucciones por medio (que, aunque son un avance respecto al simple fichero de texto, la interacción con el ratón que permiten es muy rudimentaria y para colmo se hacen insoportablemente lentas) entro al fin en la opción de dibujo. Llegado este momento, doy gracias a los dioses de que no se me cruzasen hace meses un par de cables y comprase el ratón a 7500 pesetas, porque de haberlo hecho así, y tras contemplar el penoso espectáculo que mi cerebro estaba a punto de soportar, hubiese tomado el cable del ratón y me lo hubiese puesto alrededor del cuello, colgándolo seguidamente de la lámpara del techo y dejándome caer con dulzura, observándo mi QL por última vez...
...pero afortunadamente aquella cosa que se llamaba Star Mouse sólo me había costado 1500 pesetas más gastos de envío (¡1500 pesetas más gastos de envío...! snif, snif...) y aquello me ayudó a contenerme de realizar tan dramático acto. No sigamos por el final, sino que volvamos al momento en que, con la inocencia de un ser humano que no conoce el Star Mouse, me encuentro ratón en mano presto a trabajar con mi nuevo y flamante programa de dibujo.
En cuanto hago los primeros movimientos me doy cuenta de que los autores no se quisieron o no pudieron romperse demasiado la cabeza, hecho bastante frecuente por aquí hasta hace muy poco, y no me voy a meter más en ese tema porque sería un atrevimiento hacerlo sin un mayor conocimiento de las causas. Antes de pasar a relatar todas y cada una de las excelencias observadas por mí en el programa, hay que volver a recordar que éste apareció en el mercado hace casi tres años, y que quizá ello nos pueda hacer ser más comprensivos en nuestros juicios. No obstante el Star Mouse tiene algunas limitaciones tan grandes que la fecha de su aparición no le supone excusa alguna. Hay fallos tan garrafales, que lo son en el año 1988 tanto como en el 1985 o en el 1588, aunque por entonces los antecesores de Clive estuviesen dispersando a la Armada Invencible a bordo del barco de Drake, y los únicos ratones que había eran de los que se dedicaban a zamparse las galletas de la bodega... que época tan feliz, sin Star Mouse ni nada parecido.
Como introducción, baste decir que el programa principal está escrito en SuperBASIC, y que sólo llama a algunas rutinas en Código Máquina para mover el cursor en respuesta al ratón o para realizar los rellenos de figuras. No lo he examinado linea a linea, luego puede que emplee el Código Máquina para algo más que yo no haya advertido. Hay un detalle curioso, y es que la dirección del Código Máquina parece no ser reubicable, pero el SuperBASIC no se encarga de comprobarla. Esto quiere decir que, si bien la dirección de ubicación del código se haya con un RESPR, el programa supone que siempre será la misma. De este modo, si tienes algún tipo de ampliación de memoria puedes ir quitándola o anulándola con el comando correspondiente. Lógicamente, he intentado modificar el listado BASIC de modo que valga cualquier localización del código añadiendo una variable offset a todos los parámetros de los CALL, PEEK o POKE, pero se conoce que las rutinas no son reubicables ¡Arg...! En fin, al final desistí y me limité a modificar el boot de modo que hiciese un RES_128 para anular la ampliación de memoria de la Trump Card si la dirección devuelta por el RESPR no es la esperada, que creo recordar que es 219136. ¿Alguien sabe otra solución más elegante? Si es así, le ruego que me la comunique... por curiosidad sólamente, claro, pues no creo que vuelva a cargar el Star Mouse jamás.
Nos hemos desviado del tema, de modo que allá vamos... a relatar de una vez las muchas "excelencias" del programa. Los que sufran del corazón y no tengan el Star Mouse (porque los que, sufriendo del corazón, hayan visto alguna vez eso, me temo que sería lo último que hiciesen) les ruego que no sigan leyendo, por bien de su salud.

La primera decepción que me produjo el programa (aunque ya me lo temía) es el hecho de que no permite trabajar en el modo 4. No es esta una limitación muy grave y no me voy a detener más en ella. Baste sólo decir que para cierto tipo de dibujo hubiera sido interesante disponer de dicha facilidad.
La segunda gran decepción (y esta si que es gravísima) es que no permite trabajar con la pantalla completa. Una vez pasada la impresíon de ver los iconos (que por otro lado son bastante simplones) y una pantalla tipo Apple McIntosh (salvando las distancias), uno cae en la cuenta de que no hay ningun icono que permita cambiar los iconos de lugar o desplazar el area de dibujo (como si fuese una ventana), sino que "esto es lo que hay". Es decir, que sólo se puede dibujar en un recuadro de la pantalla, donde no hay iconos, un rectángulo que en horizontal es el 0.83 del ancho total de la pantalla y en vertical el 0.72 del alto total. Quizá los autores pensaron que la principal utilidad del programa sería el diseño de postales.

Hay que alabar, aunque poquito, el honrado intento por crear un entorno tipo GEM. Digo poquito porque el resultado es algo deprimente que le hace a uno preguntarse a viva voz: ¡¿Pará qué diablos se necesita un ratón en un programa como este?! Con esto quiero decir que el ratoncito lo único que hace es estorbar y complicar, habiendo muchas operaciones de selección de opciones que serían mucho más rápidas de hacer pulsando una tecla. Para lo único que sirve el ratón es para el dibujo a mano alzada, donde, por muy malo que sea el trasto (como es el caso), siempre es algo más cómodo que emplear el teclado. Además, el caprichoso ratón parece dotado de un peculiar tic nervioso o hipo electrónico pues, bien por el hardware en sí o por la rutina en Código Máquina que lo lee, el movimiento del cursor es a veces deseperantemente impreciso e irregular, y no precisamente a causa de la superficie en que lo emplees.
Para aquellos escasos afortunados que no hayan tenido la desdicha de toparse con el Star Mouse, les describiré por encima cómo está distribuida la pantalla en el programa: la "extensa" zona de dibujo de la que hemos hablado se encuentra situada desplazada arriba y a la derecha. La columna que queda a la izquierda de la pantalla está ocupada por varios iconos, así como ocho recuadros correspondientes a los colores disponibles. La zona inferior de la pantalla incluye 26 recuadros con diferentes tramas, incluidos los ochos colores también (lo cual hace sospechar que las tramas "reales" no pueden cambiar de color, como luego veremos). Por último, el recuadro que queda en la esquina inferior izquierda está ocupada por muestras de cuatro lineas con diferentes grosores y un indicador de qué trazo está seleccionado.

La principal chapuza a destacar es que para cambiar de función o modo hay que salir antes del modo en el estemos, en lugar de poder ir directamente al icono correpondiente al nuevo modo en que queramos entrar y pulsar el botón. Para salir de un modo hay que situarse sobre una DIMINUTA BARRITA que está sobre el resto de los iconos, en la zona superior izquierda de la pantalla, y pulsar el botón, operación que por sí misma hace el uso del programa una auténtica tortura capaz de volver loco al más templado o cuando menos provocar deseos de emplear el ratón como unas boleadoras de gaucho de la Pampa argentina y arrojarlo contra el monitor una y otra vez hasta destrozarlo.
La segunda principal chapuza es que en todos los modos es necesario salir de éstos para elegir un nuevo color. El tener que ir a localizar la odiosa barrita, pulsar el botón, elegir un nuevo color situándonos sobre el recuadro correspondiente, volver a entrar en el modo en que estábamos... el hacer esto, repito, unas cuantas veces, ¡y todo para cambiar simplemente de color!, es como para agarrar el QL y emplearlo como un ariete de mano para cargarte la pantalla del monitor con tal de hacer desaparecer de tu vista el programa de marras. Existe una honrosa excepción a esto: en el modo de rellenos se puede cambiar de tipo de relleno sin salir del modo ¡Aleluya!... honrosa excepción que se desluce por otro motivos de los que me ocuparé más tarde.
En el modo en el que entramos al iniciarse el programa podemos pintar puntos a cada pulsación de botón, allí donde se halle el cursor. Esto será de utilidad para realizar circunferencias, elipses o lineas. Si entramos en el modo de dibujo a mano alzada (el icono "brocha") podremos hacer que el cursor vaya dejando su rastro allá por donde pase tras haber pulsado el botón. Para que el cursor no deje rastro podemos pulsar de nuevo el botón. El trazo será en el grueso y color seleccionados... pero si el color es el blanco el trazo será multicolor! Eso al principio puede parecer interesante, pero hace sospechar que no se puede pintar de blanco sobre otro color, claro. No sólamente eso, sino que uno se da con un canto en los dientes cuando comprueba que apenas hay nada que pueda hacerse sobre un fondo pintado de otro color...! Más que para el diseño de postales en general, los autores debieron escribir el programa para el diseño de postales de estaciones de esquí, que creo es una aplicación muy extendida entre los usuarios de QL que viven en Transilvania.
Me decido a hacer una figurita a base de lineas (¡Hay un icono para hacer lineas rectas!, ¿Cómo no hay otro para hacer pixeles redondos?) o bien a base de circunferencias y me atrevo a rellenarlo: salgo de modo, entro en el modo de rellenar (icono estilo "rodillo"), me dirijo al centro de la figura y apreto el botón... ¡Pardiez, pero si funciona! Cambio de tipo de relleno y me dirijo a otra zona... ¡Funciona! Un merecido paréntesis para dejar constancia de que la rutina de rellenar funciona a la perfección... por ahora. Cerrado el paréntesis, continúo: en primer lugar, me dediqué malsanamente a buscar como un desesperado los fallos posibles y en segundo lugar me dediqué a anotarlos. He aquí los fallos más destacables y chapuceros:
No se pueden rellenar superficies que no sean blancas. Cualquier intento de hacer esto pinta una linea horizontal dentro de la figura o, en el mejor de los casos, dos o tres. ¿A quién se le ocurre recolorear un círculo amarillo de azul?

No se pueden rellenar figuras que no sean negras o azules ¿A qué ingrato ser se le ocurriría querer colorear una circunferencia verde de color verde para hacer un círculo verde? Si dibujamos una circunferencia o cualquier polígono cerrado de negro o azul oscuro no habrá problema en rellenarlos con cualquier color, excepto blanco (¡ ¿pará que existe la posibilidad de rellenar de blanco, si sólo se pueden rellenar cosas blancas?!) Pero si la figura está dibujada con trazos de color rojo o magenta, entonces cualquier intento por rellenarla se parará antes de acabar, dejándonos con una sola franza coloreada. Si rellenamos las zonas que quedan en blanco pasará lo mismo, de modo que rellenar algo en esas condiciones es desesperante (¿será una utilidad oculta del programa, o será otra chapuza?) Si tenemos la desgracia de querer colorear una figura que tiene sus lineas o curvas de color verde, cyan, o amarillo (¿para qué vamos a hablar del blanco, verdad?) entonces se te coloreará TODA la pantalla, a menos que haya algo negro o azul que sirva de límite. ¿Original, no es cierto? Para colmo, de entre las diferentes tramas disponibles podemos escoger la que queramos, pero siempre se realizará el relleno con dicha trama en color negro, tengamos la tinta que tengamos. Hay una trama, por ejemplo, que da el aspecto de una pared de ladrillos... ¿a qué idiota se le ocurriría tratar de usarla con tinta roja para rellenar una fachada? Claro, queda mejor en color negro sobre blanco, por supuesto.
En los programas de dibujo DE VERDAD el relleno de zonas permite elegir el color o colores del límite de la zona, o bien rellena toda la zona circundante que esté en el mismo color que el pixel sobre el cual hemos activado el relleno... but Spain is still different.
Otra solemne chapuza del programa: no se puede borrar la última operación. De hecho, ¿quién iba a querer borrar la última operación si a las dos cosas que haces apagas el ordenador desesperado? En serio, ¿es que el QL no dipone de 32 Kbytes para guardar una copia de la pantalla a cada paso que se efectúa en el dibujo? El programa BASIC ocupa 8 Kbytes. El Código Máquina ocupa la sorprendente cifra de 43 Kbytes, por lo que me permito sospechar que en ese mismo espacio se encuentra la zona de almacenamiento de datos de los FILLS (rellenos de zonas), que supongo ocupan bastante. Mi impresión es que no es imposible tener una copia de la pantalla en memoria, y por ello he de concluir en que los programadores no creyeron necesario dotarnos de tal facilidad porque supusieron que nunca nos equivocaríamos al dibujar postales de estaciones de esquí, que las dibujaríamos de un tirón. Existe un icono en forma de cubo de basura que lo que hace es BORRAR TODO, dejándonos con una flamante pantalla en blanco, como recién cargado el programa. Muy útil, ¿no es así?
Tenemos otra opción de borrar, el icono en forma de presunta goma de borrar, que es para partirse de risa: cuando lo activamos el cursor se convierte en un par de lineas paralelas y al mover el ratón VAMOS PINTANDO DE BLANCO por donde pasamos. No podía ser de otra forma, pero, ¿no hubiera sido mejor poder pintar de blanco con la opción de pintar de blanco?
Otra chapuza: el movimiento de la zona de dibujo. Tras entrar en el modo de "scroll", y pulsando el botón sobre las flechas, podremos desplazar el dibujo dentro de la zona de dibujo, pero, como era de esperar, al mover el dibujo se va perdiendo por los bordes, de modo que al moverlo en sentido opuesto recuperamos una hermosa zona de fondo blanco. Esto es así porque el movimiento se ha hecho desde BASIC.
Una deficiencia bastante notable, aunque dado el nivel del programa no se echa en falta, es el tratamiento de ventanas. No se pueden definir ventanas o recuadros para moverlos, rotarlos, ampliarlos... No se pueden pedir peras al olmo. Ya digo que esto no es muy significativo, y es más: ¿a quién no le hubiera dado un soponcio si después de tanta chapuza se pudiesen hacer esas cosas?
Hay un detalle en el dibujo a mano alzada que no he comentado, pero que más de uno lo habrá dado por supuesto a estas alturas: no se puede cambiar el grosor del tramo mientras estamos dibujando, ¿alguien esperaba otra cosa? Todavía hay más: el cambio de grosor no afectará a las circunferencias o elipses, claro. ¿Quién puede tener interés en dibujar una circunferencia con un trazo grueso, es más, quién puede tener interés en dibujar una circunferencia con este programa?
Como detalle curioso, existen, dentro de la zona de dibujo, en la parte superior de ésta, unas rayitas horizontates y un rectangulito muy fardones ambos, estilo ventana de entorno GEM, que aparentemente no sirven para nada. De hecho no se puede desplazar el cursor sobre ellos para dibujar... PERO PUEDEN RELLENARSE AL COLOREAR CON UNA TRAMA UNA ZONA QUE LIMITE CON LA PARTE SUPERIOR DEL AREA DE DIBUJO. Esto da una idea del grado de "chapucería" en el que estamos inmersos.
Otro aspecto destacable, por chapucero, del programa, es el almacenamiento y recuperación de pantallas. A menos que uno modifique el BASIC (y puedo asegurar que yo no malgastaré esfuerzo alguno en ello) sólo se puede grabar un fichero de pantalla, con un nombre fijo, en cada microdrive. Vergonzoso. ¿Tan complejo es

incluir un INPUT para pedir el nombre con que se quiere grabar la pantalla? ¿Es que en cada microdrive sólo caben 32 Kbytes? (A lo peor en los de 1985 pasaba eso). Para colmo, si se produce un error de lectura o grabación el programa se queda colgado, como si se intenta detener la ejecución en cualquier punto con CTRL y la barra espaciadora.
Existe una opción de copia por impresora, que lógicamente se hace en Código Máquina, pero que carece de posibilidad alguna de modificación o adaptación así como documentación en las instrucciones. Vamos, como si todos tuviesemos la misma impresora, que por otra parte no sé cual se supone que es. Así que, si la impresión no sale, o sale a franjas, o cualquier otra cosa, mejor abandona e imprime con algún otro programa más "serio".
Otro detalle bastante decepcionate es el hecho de que no existe indicador de la posición en coordenadas del cursor, lo cual imposibilita cualquier intento de hacer dibujo lineal (y más dada la caprichosa forma en que hay que dibujar las rectas)

Hay un sólo punto que cabe la pena destacar por su calidad. SI, ES CIERTO, AUNQUE PAREZCA INCREIBLE, HAY ALGO INTERESANTE. Se trata de la creación de elipses, que, por medio de los iconos en forma de flechas, se hace muy cómoda: se puede alterar el tamaño de ambos ejes y la orientación, hasta dejarla como queramos. DE MODO QUE NO LO DUDEIS: SI ALGUIEN SIENTE UNA NECESIDAD REPENTINA DE DIBUJAR UNA ELIPSE, QUE CARGUE EL STAR MOUSE Y LO HAGA, SE SENTIRA MUCHO MEJOR. (No se puede decir entonces que el programa sea totalmente inútil, pues sería una injusticia)

A los pocos que no tengáis el programa y el escarabajo pelotero que lo acompaña, os doy mi enhorabuena más efusiva porque no sabéis de lo que os habéis librado; espero que continúe tan buena racha. A los demás, seres desafortunados y abandonados por el destino, no os desaniméis. Cuando recibáis una visita inoportuna acompañada de sobrinito y todo, poned a éste frente al QL y dejadle haciendo garabatos ante la pantalla para tenerle ocupado un buen rato y así evitar que os destroce el resto de la casa. Ahora bien, es importante que el niño no sobrepase la edad de tres años, pues si tiene al menos tres años se sentirá tan decepcionado, por el hecho de que el programa no le permite dibujar como él sabe hacerlo, que posiblemente le corte el cable al ratón con unas tijeras y lo arroje contra la pantalla tras haber arrancado a mordiscos las teclas del QL o, lo que es más grave, que no os vuelva a dirigir la palabra de por vida.

Tengo una de las baterías "semi-caseras" que vende VALENTE para mantener el reloj del QL. En algunas ocasiones (más de las que serían normales) se resetea el reloj en el encendido o apagado del ordenador. Sin embargo, con el escarabajo pelotero conectado lo más probable es encontrarse con que la flecha no se mantiene en la mayoría de los casos. Lo aviso por si alguien se encuentra en el mismo caso y porque antes de desatornillar el QL para abrirlo y comprobar si la batería se ha desoldado (como casi estuve a punto de hacer) es preferible probar a desconectar lo que todos sabemos... y de paso guardarlo en el fondo de un cajón, cerrarlo con llave y tirar ésta por la ventana.

¿Alguien sabría cómo modificar el escarabajo pelotero para hacerle simular las teclas del cursor? Quizá así sirva de algo y me anime a volver a conectarlo alguna vez. De ese modo incluso, cableando por aquí y por allá, podría conectarlo a un Spectrum; el requisito es que el ratón actué a modo de interruptor, según gira la bola, entre la patilla común del conector de joystick y las otras cinco patillas. No sé si sería sencillo de hacer, de modo que aquí dejo el guante por si alguien más ducho en hard que yo lo recoje.

Marcos Cruz (QLave-242)
Madrid, Julio 1988

COMENTARIO PROGRAMAS

PROGRAMA : TURBO XCHANGE v3.87
AUTOR : PSION
DISTRIBUIDOR : Normalmente este programa se regala al comprar un THOR.

La copia que he tenido el gusto de probar, es una copia que ha sido modificada ilegalmente para correr en un QL con 640K de Ram y unidades de disco, como minimo. Para conseguir una copia solamente hay dos caminos: un amigo te pasa una

copia, o comprar una copia al pirata de SPEM.

El programa que configura la impresora está compilado con el Qliberator y no corre en los QLs. Solución : usar el install inglés v2.3 o 2.35 (la versión v2.21 Española no es compatible).

Para usar la teclas F6 - F10 que no vienen en el teclado de un QL Standard, hay que pulsar Caps + F1 - F5.

Aunque en la propaganda se decia que los programas corrian en multitarea, no es verdad, pues en realidad solamente hay un solo programa que es multitarea con el Superbasic u otros programas. Ese programa comienza con el menu del XCHANGE TASK CONTROL que consiste en selecionar tareas (Quill, Abacus, Archive y Easel). Me atreveria a decir sin equivocarme demasiado que este programa es una especie de programa gigante con varios programas integrados.
Los comandos del Xchange Control Task son : backup, list, quit, tsl, continue, new, rename, xchange, delete, output, set.
Se muestran los dispositivos por defecto (p.e.: Flp1_ deafult y help). También Printer Status.

En QUILL hay varias cosillas nuevas, sin contar que es el doble de rápido que nuestras versiones, por ejemplo Extract es un comando que sirve para extraer un trozo de texto de un documento y guardarlo en un fichero (esto no se podia hacer con nuestras versiones). También está la opción GLOSSARY pulsando F5 que guarda un juego de teclas pulsadas en una sola tecla para su posterior uso.

Abacus ofrece nuevos comandos incluyendo protección de celdas.
Archive ofrece el comando Sedit y una función llamada USR que permite linker rutinas en código máquina, pero como no tengo instrucciones, no sé ni como funciona.
Easel incluye los famosos gráficos en tres dimensiones.

Para pasar de una tarea a otra hay que pulsar F6 para pasar al menu Xchange Task control, una ver alli seleccionar la tarea que deseamos. Las tareas se encuentran suspendidas y volvemos a encontrarla igual que cuando se abandonaron. Si volcamos un documento Quill por la impresora desde una tarea Quill, no podemos volver al menu Xchange Task control hasta que termine de imprimir, pero si podemos pasar al Superbasic u otro programa en multitarea (recordad que el Turbo Xchange es un solamente un programa gigante).

Existe un nuevo lenguaje llamado TSL (Task Sequencing Language).

El Turbo Xchange crea una Ramdisk de 473 sectores donde guarda dos ficheros, Psion_xch y y_temp. En mi QL 896K deja al comenzar unos 240 K libres.

En resumen, es una versión que corre a una velocidad impensable para los usuarios de QL, y es altamente recomendable hacerse con una copia, pues vale la pena. El único defecto que he encontrado es que el QL se queda seco de memoria al cargar ese programa, especialmente en un QL 640K.

Salvador Merino
Fuengirola (QLave-154)

PROGRAMA: LIGHTNING
AUTOR: STEVE SUTTON
EDITORIAL: DIGITAL PRECISION
PRECIO: 29.95 libras +5% de gastos de envío

Supongo que sereis pocos los que no hayais oido hablar del Lightning, y menos los que no hayais leido los comentarios al respecto en el QL World. por si acaso, baste decir que se trata de un programa que sirve para acelerar la salida por pantalla del QL, y algo más que luego veremos, interceptando las llamadas a las rutinas correspondientes de la ROM y ejecutando en su lugar las suyas propias. Lo importante es que dicho proceso se realiza de forma transparente, es decir, que el usuario no se da cuenta de ello.
Respecto a la velocidad general del QL, hay que decir que a pesar -y en algún

aspecto debido a ello- del microprocesador 68008, la máquina es lenta, comparándola con otras basadas en el 68000. Pero la razón principal de la lentitud es sin lugar a dudas el contenido de la ROM. De todos es conocida la velocidad a que fue implementado y escrito el sistema operativo y la cantidad de especificaciones que, habiendo sido anunciadas por Sinclair, se tuvieron que deshechar a última hora por problemas de espacio. Ese es el problema principal: los escasísimos 48 Kbytes que debían contener el QDOS y el SuperBasic. Tal vez hubiera sido más práctico el dedicar todo el espacio para el QDOS y proporcionar el SuperBasic en microdrive, en cuyo caso nos hubiera quedado menos espacio para programas en SuperBasic (lo cual no es un gran problema) pero dispondríamos de un QDOS con todas las facilidades del Tollkit II, por ejemplo, y mucho más rápido. Pero el caso es que a la hora de escribir el contenido de la ROM, los programadores tuvieron que hacerlo lo más compacto posible, sacrificando por ello la velocidad.

Esta oposición constante que se da al programar en código máquina (y en realidad en cualquier lenguaje, aunque tal vez en código máquina se aprecia mejor) entre espacio ocupado y velocidad de ejecución, se manifiesta mejor en lo relativo a las rutinas de impresión en pantalla.

El QL permite, en cuanto a impresión de textos se refiere, una enorme flexibilidad en tamaños y colores. La consecuencia de esto es que hay tamaños de letras y combinaciones de colores que son más sencillas de tratar (y por tanto más rápidas). Pero la pega es que el QDOS no distingue esos casos, sino que emplea el mismo código para imprimir cualquier tipo de texto.

Bueno, pasamos a comentar el Lightning. El programa lo recibí en disco de 3.5 pulgadas y 1440 sectores, como lo había pedido, junto con un manual de 20 hojas azuladas.

La instalación del programa es muy sencilla, y en cuanto se leen las primeras páginas del manual se puede hacer sin problemas. Conviene antes de nada sacar una copia del contenido del disco y trabajar con ella.

Lo que el disco contiene es un programa en Basic que se encarga de instalarnos en otro disco cualquiera el Lightning, según nosotros vayamos seleccionando qué partes del mismo queremos instalar. Lo que hace este programa configuradar es crear un fichero boot en el disco nuevo, que contenga las instrucciones de carga LBYTES de los ficheros del Lightning que hayamos elegido y las llamadas CALL al inicio de los mismos, con el fin de instalarlos.

¿Qué ficheros contiene el Lightning? En primer lugar, el programa configurador en Basic. Después los ficheros de código correspondientes a cada tipo de "aceleración", así como los programillas Basic para instalarlos aquellos individualmente. Por último, tenemos alrededor de ¡40 juegos de caracteres nuevos! y un fichero especial para el caso de correr el programa en un ATARI ST con QL EMULATOR.

¿Qué puede hacelerar Lightning? Cada aspecto del QDOS que puede ser acelerado por Lightning tiene un fichero propio que se carga e instala con el correspondiente RESPR, LBYTES, y CALL, aunque de ello no hace falta preocuparse, pues el programa instalador de Lightning lo hace todo por nosotros. Estos ficheros y su misión son:

lng_TEXT_ext acelera la impresión de textos.

lng_TEXT_mini_ext tiene la misma función que el anterior, aunque no tiene todas sus ventajas, pero ocupa mucho menos (para el caso de QL con 128 K)

lng_GRAPH_ext acelera el dibujo de puntos, líneas y circunferencias, así como la opción FILL para rellenar lo que se dibuje. También acelera el dibujo de elipses que estén orientadas sobre los ejes de la pantalla. No acelera el dibujo de arcos (función ARC), aunque es posible que una nueva versión sí lo haga. Aparte de acelerar, corrige algunos errores de precisión que tienen las rutinas de la ROM.

lng_MATH_ext acelera las siguientes funciones matemáticas del QDOS: EXP, LN, LOG10, SIN, ASIN, COS, TAN, ATAN, COT, ACOT, SQRT.

lng_MATH_newnames_ext crea duplicados de las funciones matemáticas del QDOS, de modo que podamos usar indistintamente las originales o las nuevas, más rápidas. No acelera las funciones normales, sino que añade otras más rápidas con nombres diferentes (por ejemplo FSIN en vez de SIN, etcétera).

lng_FONT_ext añade al Basic las extensiones necesarias para emplear cualquier juego de caracteres en cualquier ventana. Es el equivalente del CHAR_USE del Toolkit II.

Todos los ficheros _ext que hemos visto tiene sus homónimos terminados en _bas, que son pequeños programas en Basic que los inicializan individualmente. Por ejemplo, al hacer LRUN flp1_lng_TEXT_bas instalamos el código que acelera la impresión de textos, y así con los demás.

También se incluyen un par de demostraciones en Basic para cronometrar ciertas rutinas de pantalla con y sin Lightning. ¡La mejor demostración es probar el

Lightning con nuestros propios programas!
Cuando ejecutamos el programa configurador, éste nos irá preguntando uno a uno por los ficheros _ext que queremos meter en el disco en el que vayamos a instalar Lightning. Por ejemplo, en un tratamiento de textos habrá que meter obligatoriamente lng_TEXT_ext y ninguno más, pues no sirven de mucho los demás, pero en un programa de dibujo técnico o CAD convendría meter lng_MATH_ext sin dudarlo un instante y tal vez lng_TEXT_ext también. Todo depende del tipo de programa que haya en el disco o que vayamos a instalar después en el disco.
Una vez hemos seleccionado qué partes de Lightning queremos instalar, se nos preguntará si el disco en el que vamos a instalarlas tiene o no un fichero boot (¡esto lo podría averiguar el propio programa configurador si se tomase la molestia, digo yo!). Si respondemos que no, simplemente creará un boot con las ordenes que hagan falta y copiará los ficheros _ext seleccionados de uno a otro disco. Si hay un boot, lo renombrará como YOUR_ORIGINAL_BOOT y al final de su propio boot meterá un LRUN "flp1_YOUR_ORIGINAL_BOOT". Este último aspecto puede perfeccionarse metiendo un par de líneas en el programa configurador, de modo que nos pregunte qué nombre queremos dar a nuestro boot antiguo; ¡¡Queda mejor "boot_Quill", por ejemplo, que "YOUR_ORIGINAL_BOOT"!! Detalles aparte, el proceso es comodísimo y si nos descuidamos nos encontraremos instalando el Lightning hasta en los discos vírgenes...
He olvidado mencionar que cada fichero _ext añade al Basic ciertas extensiones que permiten desconectar las rutinas de aceleración, dejando al QL con las rutinas originales del QDOS y su lentitud... En el caso de las rutinas matemáticas, tenemos un comando para cambiar la precisión de los cálculos a 4 bytes (valor por defecto), 2 bytes o 1 byte. Otra facilidad de la aceleración de textos es un comando para acelerar el comando MODE, evitando que realice algunas operaciones en las ventanas que en ocasiones no son necesarias.
Otras facilidades de Lightning son:
- Comando _lngZIP para variar el número de lineas de texto que lng_TEXT_ext moverá al llenarse una ventana.
- Dispositivo nulo (nul) que deshecha todo lo que se le mande, muy útil para los procesos de depuración de programas.
- Facilidad de desagüe (drain) que nos permite, mediante la pulsación de CTRL Z, mandar "a la porra" todo lo que debería salir en pantalla. Por ejemplo: hacemos un COPY UPDATES_DOC,SCR de un fichero enorme y no nos interesa lo que viene al principio. Pulsamos CTRL Z y la pantalla se quedará como bloqueada (como con CTRL F5)... pero la impresión no se detendrá en realidad, sino que es como si se desviase al dispositivo nul y se realizará a toda velocidad! Pulsando una tecla cualquiera volvemos a ver en pantalla lo que se está imprimiendo (¡¡si tardamos mucho, ya no habrá nada que ver!!). En lugar de CTRL Z puede asignarse otra tecla para esa función con el comando _lngKEYSET.
Antes de hablar de velocidades, comentaré que el manual es breve pero claro y suficiente, con instrucciones precisas de cómo instalar el programa y de como emplear los comandos. Además, se proporcionan ejemplos de cómo hacer ciertos cálculos en lugar de otros con las funciones matemáticas para acelerar aún más nuestros programas Basic, así como algunos consejos a la hora de diseñar la posición y tamaño de las ventanas para sacar el máximo partido a Lightning.
¿Qué incremento de velocidad conseguimos con Lightning? No se puede dar una cifra exacta, pues depende del tipo de programa que emplemos, de las extensiones que haya instaladas, e incluso de si tenemos o no expansión de memoria y cual. Sobre todo, depende del uso repetido y continuo que se haga de la pantalla y de las funciones matemáticas. Para haceros una idea, os describo algunas pruebas que he hecho:
1. Tiempo empleado al imprimirse las pantallas de texto y gráficos de Super Astrologer y Astronomer: ¡He perdido el papel en que anoté los resultados, cronometrados con reloj de pulsera! Seguro que lo encuentro cuando envie esto... Bueno, ni me apetece repetir las pruebas (me tiré un buen rato en ello) ni merece la pena. Puedo asegurar, de memoria, que todas las pantallas tardaban en completarse al menos un 20% menos de tiempo, más o menos, con Lightning (lng_TEXT_ext + lng_GRAPH_ext + lng_MATH_ext) que sin Lightning.
2. Tiempo empleado en 1000 sustituciones de letras en The Editor. Aquí si que se notó la diferencia: cargué un fichero de 750 líneas de texto y ordené un "1000e/a/x/" para que me cambiase las 1000 primeras "aes" por "equis". Sin lightning tardó alrededor de 7 minutos (el tiempo exacto estaba en el mismo papel que antes, ejém), y con Lighting (lng_TEXT_ext) tardó alrededor de 4 minutos.
3. En el fichero forth_scr del Forth de Computer One viene una palabra para dibujar un enorme globo a base de circunferencias. La modifiqué un poco para realizar el test: 100 globos (cada globo se dibuja primero y luego se borra dibujándolo de nuevo encima con el color de tinta igual al del papel) sin

Lightning tardaban en dibujarse en pantalla el mismo tiempo en que se dibujaban 70 globos con Lightning (lng_GRAPH_ext). He de mencionar que la instalación del fichero lng_MATH_ext no aceleraba nada el proceso, a pesar de que en el dibujo de las circunferencias se empleaban cálculos de senos con las palabras de coma flotante del Forth (¿?). Tardaba lo mismo con lng_MATH_ext que sin él; sólo lng_GRAPH_ext aceleraba el asunto de forma sensible... misterios de la Naturaleza.

4. ¡No he podido ejecutar el programa Cosmos con el Lightning instalado, se queda colgado antes de aparecer el dibujo de presentación!

5. En nuestro lento Quill no he notado demasiado acelerón. Se nota, eso sí, que los menús de arriba se borran y dibujan más rápido, y que el scroll parece algo más ligero, pero el engendro es de por sí tan lento que no hay milagro que lo acelere, ya que no es problema sólo de salida en pantalla en este caso. Por ejemplo, la definición de bloques para borrarlos o copiarlos sigue siendo, a pesar del lng_TEXT_ext, tan tortuosa y desesperante como antaño.

6. Donde he apreciado espectacularmente la presencia invisible de Lightning es al copiar un fichero desde cualquier dispositivo a la pantalla. En estos casos, cronometrando empleando la función DATE, la media de tiempo empleado en hacer la copia con Lightning ha sido el ¡30%! del tiempo que lleva hacerla sin Lightning... ¡no da tiempo ni a leer la pantalla! ¡Cuando te acostumbras a esa velocidad y desconectas el Lightning con la orden _lngoFF, te da la impresión de que le pasa algo al QL, de lo lento que va imprimiendo!

Bueno, creo que con estas impresiones os habréis hecho una idea de las características del programa. Mi opinión particular es que merece la pena sin lugar a dudas el comprarlo, pero que hay que tener claro que donde se nota su presencia es en aquellas tareas que hagan un uso repetido y constante de la salida a pantalla o de las funciones matemáticas ya citadas.

Los juegos de caracteres incluidos en el disco del Lightning son muy variados, desde futuristas hasta medievales, aunque sólo incluyen hasta el caracter 127, es decir, ningún caracter español. con un QLUDGE como el de Turbo Toolkit o SpeedScreen se pueden modificar y, con algo de paciencia, añadirles las vocales acentuadas, las "eñes" y demás. Con el siguiente programita, adaptación personal de uno que viene en el manual del Lightning, podremos sacar en impresora unas copias de todos los juegos de caracteres de un disco (está hecho para la Trump Card, o bien para el Toolkit II). He empleado CHAR_USE del Toolkit II en lugar de la orden _lngFONT del Lightning, para evitar tener que cargar e instalar el fichero lng_FONT_ext.

Una última nota antes de acabar: si, en el plazo de un mes después de comprar el Lightning, se compra un compilador de Digital Precision (Superforth, C, Turbo) se obtiene un 15% de descuento con sólo mencionar en la carta el nombre del disco del Lightning (el nombre con que fue formateado), que es una serie de dígitos.

```
100 REMark Programa para imprimir juegos de
110 REMark caracteres con Toolkit II
120 REMark
130 REMark Marcos 24/10/1988
140 REMark
150 TK2_EXT
160 FORMAT ram1_200
170 :
180 MODE 4
190 WINDOW 480,240,0,0
200 PAPER 7
210 INK 0
220 CSIZE 1,0
230 CLS
240 :
250 SDP_SET 1,3,0,0 :REMark depende del tipo de impresora!
260 device$="flp1_"
270 OPEN_NEW#3,ram1_dir
280 WDIR#3,device$&"_font"
290 CLOSE#3
300 dire=ALCHP(1024)
310 OPEN_IN#3,ram1_dir
320 INPUT#3,basura$,basura$
330 número_fonts=0
340 REPeat print_font
350   IF EOF(#3):EXIT print_font
360   INPUT#3,nombre_font$
```

```
370   IF nombre_font$(LEN(nombre_font$)-4 TO)<>"_font":NEXT print_font
380   LBYTES device$&nombre_font$,dire
390   CHAR_USE 0,0
400   PRINT nombre_font$
410   CHAR_USE dire,0
420   FOR n=32 TO 127:PRINT CHR$(n);
430   número_fonts=número_fonts+1
440   PRINT
450   PRINT
460   IF número_fonts=6
470     hard_copy
480     número_fonts=0
490     CLS
500   END IF
510 END REPeat print_font
520 IF número_fonts:hard_copy
530 CHAR_USE 0,0
540 STOP
550 :
560 DEFine PROCedure hard_copy
570   BEEP 1000,10000
580   PAUSE(-1)
590   SDUMP#1
600 END DEFine hard_copy
610 :
```

Marcos Cruz (QLave-242)
Madrid, Noviembre 1988

COMENTARIO DE LIBROS

TITULO: Introduction to Forth
AUTOR: Ken Knecht
EDITORIAL: Howard W. Sams & Co., Inc., 1983, (USA)

Evidentemente se trata de un libro sobre Forth, pero no es un libro sobre el lenguaje Forth en general sino sobre la versión MMSFORTH del mismo, para el ordenador Radio Shack TRS-80 modelos I y III (aunque el MMSFORTH también existe para IBM PC y compatibles). Con sólo decir eso basta para entender que el interés del libro es bastante limitado. El TRS-80 es una vieja reliquia "preSinclairiana" que algunos no sabrán ni qué forma tenía. Por curiosidad, baste decir que estaba basado en un Z-80 y empleaba discos de 8 pulgadas (me parece). Entonces, ¿qué interes puede tener el libro para un usuario del QL? No tengo ni idea, pero lo tenía a mano y me ha dado por ponerme a escribir. Me temo que sólo los aficionados al Forth compraríamos algo así; yo me hice con él sin fijarme demasiado (era el único libro sobre Forth que encontré entonces) y luego me llevé la sorpresa de que no trataba de fig-Forth o Forth-83, sino sobre una cosa rara que se llamaba MMSFORTH, y además con continuas referencias al maldito TRS-80.
Lógicamente, el inicio del libro consiste en una introducción al Forth en general, lo típico en estos casos, pero a partir del tercer capítulo no hay quien se libre del TRS-80 y del MMSFORTH. Y el caso es que las peculiaridades de la "versión esta" del Forth son bastantes, de modo que para alguien que trate de iniciarse en Forth le recomiendo que se mantenga alejado de este libro, si es que alguna rara vez se encuentra con él en una librería especializada. En resumen, una rareza para Forth-maniacos.

TITULO: Lenguaje Forth para micros
AUTOR: Steve Oakey
EDITORIAL: Paraninfo, 1985
TITULO ORIG.: Forth for micros

Esto ya es otra cosa. Si alguien siente curiosidad por saber qué es esto del Forth, que adquiera este libro, que para colmo es muy fácil de encontrar. Todos

los ejemplos del libro están escritos de forma que sean ejecutables bajo MMSFORTH (¡quién lo iba a decir!), empleando el conjunto mínimo de palabras del FORTH-79 (el MMSFORTH es más amplio que el FORTH-79, según parece). Lo más interesante es que se proporcionan las modificaciones necesarias, cuando es inevitable hacerlas, para ejecutar los programas bajo fig-Forth, bajo el Forth del Dragón, ¡e incluso bajo el Forth del Jupiter Ace! El Jupiter Ace, para quien no lo sepa, era (o es) un pequeño ordenador de color blanco (parecido al antiguo ZX-80) que traía el Forth en ROM, en lugar del tradicional Basic.
El libro pasa repaso correctamente y en orden a todos los aspectos del Forth, desde qué es la pila hasta la creación de palabras para definir matrices de varias dimensiones. En algunos casos se comparan los ejemplos en Forth con sus equivalentes en Basic, lo cual será de gran ayuda a quienes "vengan" de dicho lenguaje. Un interesante anexo da información sobre qué palabras están presentes en qué versiones de Forth (Dragón Forth, FORTH-79, Jupiter Ace, MMSFORTH, fig-Forth).
Un libro interante para iniciarse o incluso para aprender algunas cosillas más los ya iniciados. Totalmente recomendable.

TITULO: La Máquina Superinteligente
AUTOR: Adrian Berry
TITULO ORIG.: The Superintelligent Machine - An Electronic Odyssey
EDITORIAL: Alianza Editorial, 1983
COLECCION: Libro de Bolsillo, número 970

No sé cómo empezar a comentar este libro, así que empezaré diciendo simplemente que es una delicia leerlo. Y digo que no sé cómo empezar porque no trata de nada que se pueda describir en un par de frases. No es un libro técnico, pero tampoco es un relato, sino que es un poco de todo. Por supuesto, el tema central es la informática, pero abarca tantos aspectos que es algo comprometido tratar de comentarlo. No obstante, lo intentaré.
Todo el libro, en el fondo, gira en torno al "problema" de la Inteligencia Artificial. La primera parte, denominada "Aprendiendo a pensar", es un conjunto de tantos temas que es muy dificil hacer un resumen correcto:
En ella se dan ejemplos de razones que durante siglos se han blandido para demostrar la imposibilidad de que el ser humano cree una "máquina" realmente inteligente y, cuando merece la pena (algunas son tan ridículas que no le merecen al autor el esfuerzo de rebatirlas), se rebaten con toda lógica y sentido común. Se insiste mucho en la distinción entre inteligencia real e inteligencia fingida, con ejemplos de todo tipo, se habla también del famosísimo test de Turing (para quien no lo conozca, esta fue una prueba que inventó el científico Alan Turing como método para saber si una máquina es o no realmente inteligente) y se hace una introducción sencilla pero efectiva al mundo de los ordenadores. También se introduce al lector en los principios de la programación (con incluso algún pequeño ejemplo en Basic). La parte más interesante y amena es aquella en que se describen las respuestas dadas por ciertos programas como el famosísimo Eliza o el SHRDLU (que mediante un brazo robot manipula bloques de colores según se le ordena) y otros similares. Se entrevista a un programa, una especie de sistema experto, creado para adoptar la personalidad de un paranóico, y se analizan sus respuestas. Del mismo modo se estudian ejemplos de cuentos o resúmenes de noticias creados por programas destinados a tales fines. Las limitaciones y los logros de dichos programas se exponen con objetividad, y se analizan por encima recursos para optimizar búsquedas y para analizar frases sintácticamente. Un par de capítulos tratan del reconocimiento de formas y del reconocimiento de sonidos, puntos cruciales en la Inteligencia Artificial. La primera parte del libro concluye con un capítulo en el que se aventuran las características y arquitectura de un posible superordenador futuro.
La segunda parte del libro, titulada "Hacia el futuro lejano", describe ciertas posibilidades relacionadas con los ordenadores, pero mirando mucho más lejos en el tiempo:
En esta parte se habla de las máquinas de Von Neumann (John Von Neumann estableció en 1948 los principios teóricos para lograr máquinas que construyesen réplicas de sí mismas), y se aplica su principio para describir hipotéticas máquinas que fuesen autosuficientes y pudiesen desplazarse de una a otra estrella, construyéndo copias de sí mismas a base de los materiales de los planetas que encontrasen a su paso. Todavía más lejos en el futuro, se formula la teoría de la existencia de supermáquinas inteligentes que fuesen capaces de llevar a cabo la gigantesca ingeniería que sería precisa para detener la

expansión o contracción del Universo...
Siento haber desglosado tanto el libro, pero no hay otra forma posible de comentar de qué trata. Todo el estilo de la obra es muy ágil, con muchos ejemplos, anécdotas, relatos históricos, metáforas... Por supuesto ninguno de los temas mencionados se trata con gran profundidad, pues no es ese su objetivo. Para aficionados a la informática en general o a la Inteligencia Artificial en particular, el libro es una auténtica delicia.

Marcos Cruz (QLave-242)
Madrid, Julio 1988

LET'S GO FORTH

Capítulo primero y espero que no el último:
"SISIFOrth, o como dejar en evidencia a los demás lenguajes"

He de comunicar antes de nada que mi solicitud de ingreso en FIG, amablemente proporcionada por Salvador Merino, ha emprendido recientemente camino aéreo hacia California de modo inevitable.
Mientras tanto, vamos a ocuparnos de nuestro tan minoritario Forth -somos pocos, pero estamos muy locos-, el mejor lenguaje que se ha inventado en esta zona del Universo conocido y en parte del desconocido.
Con el fin de intentar mostrar a los escépticos la superioridad del Forth sobre el Basic, especialmente en su ampliabilidad, flexibilidad, ahorro de espacio, velocidad de ejecución, etcétera, etcétera, he desempolvado un antiguo (2 añitos, no más) programita en Forth que escribí para Spectrum. El compilador empleado fue el de Melbourne House, que estaba aceptado por el FIG y por tanto era fig-Forth. Para adaptarlo al Forth-83 de Computer One en el QL, he tenido que cambiar un par de cosillas apenas y de paso lo he perfeccionado un poco.
Se trata de un sencillo pero original programa llamado SISIFOrth (un poco de mitología le sacará de dudas a más de uno respecto al nombre) que permite crear sistemas "inexpertos" sobre cualquier tema de forma muy sencilla. Digo "inexpertos" porque los sistemas que pueden crearse no son demasiado sofisticados en principio, ya que el programa en sí no fue hecho para eso "en serio" sino para experimentar un poco con las posibilidades del Forth, pero el hecho de que se pueda hacer lo que luego veréis (y comprobaréis si tecleáis un poquito) en tan poco espacio y de forma tan concisa y flexible demuestra sin lugar a dudas la superioridad clara del Forth sobre muchos otros lenguajes, por no decir todos. No obstante, y aunque los ejemplos de sistemas expertos que mostraremos más tarde son simplemente ejemplos rudimentarios, con un poco de paciencia podría elaborarse gracias a SISIFOrth un sistema experto bastante bueno sin alterar a SISIFOrth en sí, sino sólo con emplear adecuadamente las herramientas que él pone en nuestras manos.

* LISTADO DEL CODIGO FUENTE DEL ENTORNO SISIFORTH *

A continuación aparecen todas y cada una de las palabras del sistema, tal y como deben incluirse en una pantalla Forth. Como se verá, la cantidad de código de SISIFOrth es mínima... ¿alguien puede imaginar cuánto ocuparía un programa en Basic que hiciese esto mismo que hace SISIFOrth? ¡¡El núcleo de SISIFOrth ocupa 1 Kbyte de código fuente, que compilado ocupa apenas medio kbyte!! ¿Qué otro lenguaje puede ofrecer tanta flexibilidad y eficacia?
Nota: Por deferencia para con el autor, agradeceríamos respetasen los títulos de crédito del texto fuente, GRACIAS.

```
( Entorno SISIFOrth V2.00, Por: Marcos Cruz, 11/1988)
: TASK ; 128 CONSTANT PREGUNTAS VARIABLE HECHAS PREGUNTAS ALLOT
VARIABLE N& VARIABLE NP : .TITULO CR
." Entorno SISIFOrth V2.00 " .CPU ; : PREPARA_SISIFORTH HECHAS
PREGUNTAS ERASE CLS .TITULO CR ." Por Marcos Cruz 1986, 1988"
CR ; : S/N CR BEGIN KEY DUP DUP 83 = SWAP 78 = OR NOT WHILE DROP
 REPEAT 83 = DUP IF ." >Si" ELSE ." >No" ENDIF ABS CR ;
: PREGUNTA CREATE C, C, 1 NP DUP @ , +! DOES> DUP 2+ @ DUP
HECHAS + C@ ?DUP IF 1 - ROT ROT 2DROP ELSE SWAP DUP C@
SWAP 1+ C@ CR .LINE S/N DUP 1+ ROT HECHAS + C! ENDIF ;
: (( 0 N& ! ; IMMEDIATE : & 1 N& +! [COMPILE] IF ; IMMEDIATE
: ())) IF CR SWAP .LINE CR QUIT ELSE 2DROP ENDIF ;
```

```
: )) COMPILE ())) N& @ ?DUP IF 0 DO [COMPILE] ENDIF LOOP ENDIF
; IMMEDIATE : BLK+ 32 WORD NUMBER DROP +BLOCK STATE @ IF
[COMPILE] LITERAL ENDIF ; IMMEDIATE .TITULO .( compilado) CR
;S
```

* COMO IMPLEMENTAR UN SISTEMA EXPERTO EN EL ENTORNO SISIFORTH *

Lo que en realidad forman todas las palabras anteriores no es en sí un programa para hacer sistemas expertos, sino un conjunto de herramientas para hacerlos, una especie de sublenguaje. Una vez que dichas palabras han sido compiladas disponemos de las herramientas necesarias para definir preguntas, reglas y soluciones.

Entendemos por regla el conjunto de las condiciones que, de cumplirse todas según ciertas normas, dan lugar a una solución. Cada condición no es más que una pregunta que se hace al usuario del sistema experto. Las condiciones vendrían a ser los síntomas, y la solución sería el diagnóstico. El sistema experto toma cada una de sus reglas y prueba a verificarlas haciendo las preguntas pertinentes.

Es importante citar que el orden en que luego el programa intente probar cada regla es muy importante, pues en el momento en el que descubra que una de ellas se cumple, se detendrá y no hará caso de las restantes. Por ello hay que tener cuidado de situar las reglas que tengan más condiciones las primeras, y las que tengan menos condiciones las últimas.

Voy a poner un ejemplo, para que quede esto del orden de las reglas perfectamente entendido: supongamos que estamos intentando hacer un sistema experto de enfermedades exóticas. Suponemos que dos de las reglas son las siguientes:

- Si el paciente tiene manchas amarillas en la nariz y hecha espuma por la boca: tiene fiebre tifoidea paletoide.
- Si el paciente tiene manchas amarillas en la nariz y hecha espuma por la boca y delira: tiene parálisis nasal tropical.

Dadas esas dos reglas, en ese orden, el programa empieza por intentar demostrar la primera preguntando por los dos síntomas que la definen. Si se cumplen, el programa se detiene sin más... ¡pero la siguiente regla podría aún cumplirse, pues consta de los síntomas de la primera y uno más, que no ha sido preguntado! Por ello, el orden debería ser el inverso. ¿Entendido? No obstante, hay algunos casos en que ciertas reglas especiales, aunque tengan menos condiciones, deben ir al principio. Por ejemplo la siguiente regla:

- Si no respira y no tiene pulso: ya no hay nada que hacer.

Esa regla debería ir la primera de nuestro sistema experto de enfermedades, pues, si no, empezaría a preguntar síntomas sin tener en cuenta que el paciente esté vivo.

Una vez aclarado lo anterior, ¿cómo escribir las reglas en nuestro sistema? Primero hay que incluir la lista de las preguntas que necesitaremos hacer para todas las reglas, en el formato siguiente:

np nl PREGUNTA nombre_pregunta

Donde:

np = número de la pantalla en que está el texto de la pregunta

nl = número de linea de esa pantalla en que está el texto de la pregunta

nombre_pregunta = nombre de la pregunta, para referirnos a ella después en las reglas

El entorno SISIFOrth se encarga de organizarse a sí mismo y llevar la cuenta de las preguntas que se van creando y demás.

Después hay que definir las reglas, que son palabras Forth, en el formato siguiente:

: nombre_regla ((nombre_pregunta1 & nombre_pregunta2)) ;

Donde:

nombre_pregunta1, nombre_pregunta2 = nombres de la pregunta que hay que realizar, correspondientes a las condiciones que deben cumplirse.

Puede haber en cada regla desde una sola pregunta hasta todas las que se necesiten para que la regla se cumpla, teniendo en cuenta que entre cada dos nombres de preguntas siempre debe ponerse la palabra &.

Además, es posible, si, para que la regla se cumpla, la respuesta a alguna pregunta debe ser negativa, añadir un NOT detrás de su nombre. Por ejemplo:

: nombre_regla ((nombre_pregunta1 & nombre_pregunta2 NOT & nombre_pregunta3)) ;

Esta última regla se cumpliría si la respuesta a la pregunta nombre_pregunta1 fuese afirmativa, la de la respuesta nombre_respuesta2 fuese negativa, y la de la respuesta nombre_respuesta3 fuese afirmativa. Si alguna de las tres respuestas no fuese como se ha dicho, SISIFOrth renunciaría a intentar hacer más preguntas de esa regla y pasaría a la siguiente, si la hay.

También podría emplearse la función OR para "orear" dos o más respuestas:
: nombre_regla ((nombre_pregunta1 nombre_pregunta2 OR)) ;
Lo que ya no es tan recomendable es emplear AND, pues si hacemos...
: nombre_regla ((nombre_pregunta1 nombre_pregunta2 AND)) ;
...entonces siempre se harán las dos preguntas, ¡aun cuando, si la primera ha sido respondida negativamente, ya no haga falta seguir! Lo correcto es emplear la palabra & entre ambas preguntas, lo cual tiene el mismo efecto que AND pero es más inteligente y además está para eso, pues si la primera pregunta falla, se pasa a la siguiente regla. De hecho la inteligencia del sistema se basa en las palabras & y)), que compilan los IF...ENDIF, pues de otro modo el sistema siempre haría las mismas preguntas en el mismo orden y resultaría ridículamente estúpido.
Hay que aclarar que SISIFOrth nunca hace la misma pregunta dos veces. Para ello lleva la cuenta de las preguntas que ha efectuado y de qué respuesta se obtuvo anteriormente en cada una.
Tenemos las preguntas y tenemos las reglas. Ahora nos falta decir qué reglas conducen a qué respuestas y ordenar las reglas en el orden más adecuado según lo ya explicado. Esto se realiza incluyendo dentro de la definición de la palabra SISIFORTH_nombre, donde "nombre" es el nombre particular del sistema experto, las llamadas a los nombres de las reglas, según el formato:
np nl nombre_regla
Donde:
np = número de pantalla en que está el texto de la posible respuesta
nl = número de linea de dicha pantalla en que está el texto de la respuesta
nombre_regla = nombre de la regla
Detrás de todas las llamadas a las reglas, sería conveniente añadir por medio de la palabra ." un mensaje, para el caso de que ninguna de las reglas se haya verificado, caso que debería evitarse en un sistema experto medianamente serio.
Conviene aclarar, aunque se verá en los listados fuente de las implementaciones de los sistemas expertos que siguen, que en lugar de poner directamente el número de pantalla en que están los mensajes, se ha puesto la expresión BLK+ n, donde n es el desplazamiento desde la pantalla actual a la que tiene los textos. Por ejemplo, para definir una pregunta cuyo texto está en la siguiente pantalla, en la línea 3, haríamos:
BLK+ 1 3 PREGUNTA nompre_pregunta
El hacer eso así permite que el código fuente sea independiente del bloque en que se haya escrito.

* SISTEMAS EXPERTOS DE EJEMPLO *

A continuación se dan algunos ejemplos de cómo implementar sistemas expertos en el entorno SISIFOrth. Son ejemplos sencillos y rudimentarios, como ya antes he dicho. Ello no quita, como también cité antes, que no puedan crearse sistemas expertos medianamente útiles y sofisticados mediante SISIFOrth. Lo importante es ver, a través de los ejemplos siguientes, lo sencilllo que es realizar la implementación... lo difícil es diseñar en sí mismo un sistema experto sofisticado, es decir definir correctamente sus reglas, condiciones, preguntas... y en eso ni SISIFOrth ni el Forth pueden ayudar. Posiblemente intente buscar una idea para un sistema experto que pueda resultar interesante y sirva de muestra, y si alguien se siente interesado por ello que se ponga manos a la obra también.
Cada implementación de sistema experto ocupa cierto número de pantallas que deben respetarse. Para ello he metido una línea con tres asteriscos allí donde haya que seguir escribiendo el texto fuente en un nuevo bloque. Para poner en marcha cada implementación basta tener el entorno SISIFOrth compilado y después compilar la pantalla en que comience la implementación. Luego, ejecutar la palabra SISIFORTH_nombre que es la última que se define en cada sistema experto.
Pueden tenerse compiladas tantas implementaciones de sistemas expertos como se quiera, siempre, lógicamente, que no haya coincidencia entre sus respectivos nombres de preguntas y reglas, y cuidando que la constante PREGUNTAS de SISIFOrth haya sido inicializada con el número máximo de preguntas que vayan a existir en cualquier momento.
Nota: la idea (las preguntas y reglas) de los sistemas expertos en transporte y en créditos, la he sacado de la enciclopedia "mi computer", tomo 10 (página 2201), en donde se emplean para comentar y probar el paquete Expert-Ease de Thorn-EMI Software para PC y Mac, que es "también" un programa para crear sistemas expertos.

* SISIFORTH, EXPERTO EN ORDENADORES *

```
( SISIFOrth, experto en ordenadores: preguntas y reglas)
( Definición de las preguntas:)
BLK+ 2 6 PREGUNTA FORTH_MEJOR
BLK+ 2 2 PREGUNTA TENER_AMSTRAD
BLK+ 2 3 PREGUNTA TENER_ZX
BLK+ 2 4 PREGUNTA TENER_FORTH
BLK+ 2 5 PREGUNTA TENER_QL
( Definición de las reglas:)
: RATA_DE_CLOACA (( TENER_AMSTRAD )) ;
: CHICO_LISTO (( TENER_QL & FORTH_MEJOR & TENER_FORTH )) ;
: BUEN_CHICO (( TENER_ZX & FORTH_MEJOR & TENER_FORTH )) ;
: DESPISTE1 (( TENER_ZX & FORTH_MEJOR & TENER_FORTH NOT )) ;
: DESPISTE2 (( TENER_QL & FORTH_MEJOR & TENER_FORTH NOT )) ;
: NO_TENER_SINCLAIRS (( TENER_QL NOT & TENER_ZX NOT )) ;
: NO_FORTH_MEJOR (( FORTH_MEJOR NOT )) ;
-->
                   *         *         *
( SISIFOrth, experto en ordenadores: palabra principal)
: SISIFORTH_ORDENADORES PREPARA_SISIFORTH
." Ejemplo de aplicación del entorno SISIFOrth" CR
BLK+ 1 08 RATA_DE_CLOACA
BLK+ 1 09 CHICO_LISTO
BLK+ 1 10 BUEN_CHICO
BLK+ 1 11 DESPISTE1
BLK+ 1 11 DESPISTE2
BLK+ 1 08 NO_TENER_SINCLAIRS
BLK+ 1 08 NO_FORTH_MEJOR
;
;S
                   *         *         *
( SISIFOrth, experto en ordenadores: textos)
( Textos de preguntas:)
¿Tienes un ordenador o cualquier aparato Amstrad?
¿Tienes un Spectrum, o un ZX-81, o un ZX-80, o un Z-88?
¿Tienes algún compilador de FORTH para tu ordenador?
¿Tienes un magnífico QL?
¿Piensas que el FORTH es el mejor lenguaje del Universo?
( Textos de respuestas:)
Eres una rata de cloaca repugnante
Eres un tipo inteligente, llegarás muy lejos
Vas por buen camino, pero tienes que dar un salto cuántico
Andas algo despistadillo, ¿verdad? Deberías hacerte con un FORTH
```

* SISIFORTH, EXPERTO EN TRANSPORTES *

```
( SISIFOrth, experto en transporte: preguntas y reglas)
( Definición de preguntas:)
BLK+ 1 5 PREGUNTA DINERO
BLK+ 1 6 PREGUNTA PASAJEROS
BLK+ 1 7 PREGUNTA EQUIPAJE
BLK+ 1 8 PREGUNTA VELOCIDAD
BLK+ 1 9 PREGUNTA EJERCICIO
( Definición de reglas:)
: METRO (( DINERO NOT )) ;
: COCHE (( PASAJEROS EQUIPAJE OR )) ;
: MOTO (( VELOCIDAD )) ;
: C5 (( EJERCICIO NOT )) ;
: BICI (( EJERCICIO )) ;
-->
                   *         *         *
( SISIFOrth, experto en transporte: palabra principal y textos)
: SISIFORTH_TRANSPORTE PREPARA_SISIFORTH
." Sistema experto en medios de transporte" CR BLK+ 0 11 METRO
BLK+ 0 12 COCHE BLK+ 0 13 MOTO BLK+ 0 14 C5 BLK+ 0 15 BICI ; ;S
(Textos de preguntas:)
¿Tienes suficiente dinero para comprarte algún tipo de vehículo?
¿Vas a llevar pasajeros?
¿Vas a llevar equipaje?
¿Te gusta ir a toda leche, con cara de velocidad?
¿Te gusta hacer ejercicio?
```

```
( Textos de respuestas:)
Pues vete en metro, imbécil.
Entonces cómprate un coche.
Una moto es la solución ideal.
Te aconsejo que vayas en Sinclair C5.
Mejor será que te compres una bicicleta.

* SISIFORTH, EXPERTO EN CONCESION DE CREDITOS *

( SISIFOrth, experto en concesión de créditos)
BLK+ 0 9 PREGUNTA SEGURIDAD BLK+ 0 10 PREGUNTA DEUDA BLK+ 0 11
PREGUNTA GIRADO BLK+ 0 12 PREGUNTA INGRESOS : OK1 (( SEGURIDAD
)) ; : NO1 (( DEUDA )) ; : SO1 (( GIRADO & INGRESOS )) ; : NO2
(( GIRADO & INGRESOS NOT )) ; : SO2 (( GIRADO NOT & INGRESOS
NOT )) ; : OK2 (( GIRADO NOT & INGRESOS )) ; : SISIFORTH_CREDITO
 PREPARA_SISIFORTH ." Experto implementado: concesión créditos"
CR BLK+ 0 13 OK1 BLK+ 0 14 NO1 BLK+ 0 15 SO1
BLK+ 0 14 NO2 BLK+ 0 15 SO2 BLK+ 0 13 OK2 ; ;S
¿Ofrece seguridad el cliente?
¿Tiene el cliente algún antecedente de deuda irrecuperable?
¿Ha girado alguna vez el cliente en descubierto?
¿Tiene el cliente ingresos regulares comprobables?
Crédito concedido.
Crédito denegado.
Por favor, someter esta solicitud a un gestor para su estudio.
```

Marcos Cruz (QLave-242)
Madrid, Noviembre 1988

BOOT

La mayoría de los programas del QL pertenecen a una de las siguientes dos categorías principales, "extensiones residentes" y "programas transitorios". El SuperToolkit II es un ejemplo de una extensión residente, Quill lo es de un programa transitorio. Tal como su nombre indica, una extensión residente se diseña para ser cargada al principio de cada sesión de trabajo con el QL, y permanece residente por el resto de la sesión. De forma similar, un programa transitorio se carga cuando se le necesita, y puede ser eliminado de la memoria del QL cuando no se le necesite más.

Los programas invocados mediante "hotkey" son un caso especial diferente a los dos anteriores, ya que estos programas residen permanentemente, pero se empieza un programa transitorio al pulsar la "hotkey".

La situación se complica por el hecho de que algunos programas transitorios precisan de la presencia de un grupo particular de extensiones residentes, las cuales no pueden ser cargadas mientras haya programas transitorios presentes en el QL. La primera restricción es impuesta por el mismo programa, mientras que la segunda la impone el QDOS.

Este problema es suavizado por la facilidad BOOT del QL, que está pensada para ser usada en la carga de todas las extensiones residentes necesarias, las cuales pueden proceder de fuentes diferentes. El fichero BOOT se usa también en muchos programas comerciales para permitir a los usuarios el acceso inmediato al nuevo logicial - algunos usuarios no sobrepasan nunca este punto ¡ y reinicializan su QL cada vez que quieren cambiar de programa !.

La modificación de los programas BOOT para hacer frente a los nuevos logiciales, puede ir de muy fácil a imposible. Es fácil para los programas que se cargan con EXEC disN_nombredefichero, en cuyo caso no es necesario modificar tu propio fichero BOOT. Conversiones difíciles son aquellos casos en los que el fichero BOOT del programa original cae en la tentación de incluir mensajes de copyright, preciosos bordes, sonidos musicales u otros métodos de obscurecer los trozos útiles del código del fichero. Ficheros BOOT imposibles son aquellos que incluyen POKEs, o empiezan una aplicación con una sentencia CALL - estos pueden ser usados en algunas ocasiones, pero requieren el trabajo de un experto buceador en el código máquina, para poderlos convertir a una forma higiénica.

Para modificar tus propios programas BOOT, debes determinar que extensiones residentes son necesarios para ejecutar el logicial. Se puede determinar este extremo leyendo el manual o examinando el propio fichero BOOT del programa; puede pensarse que es una extensión residente cualquier código cargado por sentencias como la siguiente:

```
base=RESPR(tamaño):LBYTES disN_nombredefichero,base:CALL base
```

Puedes copiar las sentencias a tu propio fichero BOOT en el lugar apropiado, y puedes copiar también a tu disco BOOT normal el propio fichero contenedor de las extensiones. La forma anterior puede encontrarse repartida en varias líneas, u obscurecida por métodos como el de reservar solamente un área con el procedimiento RESPR y cargar mediante LBYTES varios ficheros dentro de dicha área. El fundamento, sin embargo, sigue siendo el mismo.
En los siguientes ejemplos, los tamaños dados para los ficheros no son necesariamente precisos: deberás usar el menú "Ficheros" de QRAM o el SuperToolkit II para encontrar el tamaño real necesario. Se asume que el medio BOOT es "flp1_", el cual, evidentemente, puede ser cambiado por cualquiera de tu elección. Todos los ejemplos usan la versión "ptr_gen" de la Interfase del Puntero, la cual trabaja tanto con la Interfase Interna para Ratón de QJUMP como con la Interfase del SuperMouse de Sandy, además de con el teclado. Esta versión sustituye a las versiones anteriores "ptr_kbd", "ptr_imi", etc. y a la invocada mediante el comando POINTER del SuperMouse de Sandy.

1/Un ejemplo sencillo de fichero BOOT para cargar y activar QRAM

```
100 base=RESPR(12388):LBYTES flp1_ptr_gen,base:CALL base
110 base=RESPR(7762):LBYTES flp1_wman,base:CALL base
120 base=RESPR(25882):LBYTES flp1_hotkey,base:CALL base
130 HOTKEY
```

La sentencia HOTKEY de la línea 130 empieza el programa transitorio llamado HOTKEY, el cual es el responsable de que al pulsar "ALT"+"/" se empiece QRAM. Una vez que empieza el programa HOTKEY, no es posible reservar espacio para ninguna otra extensión residente sin eliminar dicho programa HOTKEY. Por ello la sentencia HOTKEY debe ser colocada después de todas las sentencias RESPR del fichero BOOT.

2/ Incluyendo el SuperToolkit II con QRAM

```
100 base=RESPR(16384):LBYTES flp1_tk2_rext,base:CALL base
110 base=RESPR(12388):LBYTES flp1_ptr_gen,base:CALL base
120 base=RESPR(7762):LBYTES flp1_wman,base:CALL base
130 base=RESPR(25882):LBYTES flp1_hotkey,base:CALL base
140 HOTKEY
```

o bien

```
100 TK2_EXT
110 base=RESPR(12388):LBYTES flp1_ptr_gen,base:CALL base
120 base=RESPR(7762):LBYTES flp1_wman,base:CALL base
130 base=RESPR(25882):LBYTES flp1_hotkey,base:CALL base
140 HOTKEY
```

La línea 100 inicializa el SuperToolkit II, en el primer caso a partir de un fichero "tk2_rext" creado mediante la versión configurable del Toolkit, en el segundo caso a partir de la ROM de una interfase de disco convenientemente equipada.

3/ Un fichero BOOT con QRAM y QTYP juntos

```
100 base=RESPR(5424):LBYTES flp1_qtyp_qspell,base:CALL base
110 base=RESPR(12388):LBYTES flp1_ptr_gen,base:CALL base
120 base=RESPR(7762):LBYTES flp1_wman,base:CALL base
130 base=RESPR(29538):LBYTES flp1_hotkey,base:CALL base
140 HOTKEY
```

Como en el caso del SuperToolkit II, se cargan las extensiones SPELL de la forma normal: el programa QTYP propiamente dicho se supone incluido en el "flp1_hotkey" junto a QRAM.

4/ SuperToolkit II, QMON, QRAM, QTYP, QPTR y disco RAM/amortiguador de impresora

```
100 base=RESPR(16384):LBYTES flp1_tk2_rext,base:CALL base
110 base=RESPR(11242):LBYTES flp1_qmon_bin,base:CALL base
120 base=RESPR(5424):LBYTES flp1_qtyp_qspell,base:CALL base
130 base=RESPR(12388):LBYTES flp1_ptr_gen,base:CALL base
140 base=RESPR(7762):LBYTES flp1_wman,base:CALL base
150 base=RESPR(29538):LBYTES flp1_hotkey,base:CALL base
160 base=RESPR(9234):LBYTES flp1_qptr,base:CALL base
170 base=RESPR(5108):LBYTES flp1_ramprt,base:CALL base
180 HOTKEY
200 OUTLN #0;512,256,0,0
210 IF RMODE=8 THEN
220   WINDOW #0;448,40,32,216
230 ELSE
240   WINDOW #0;512,50,0,206
250 END IF
260 AT #0;1,0
```

De la forma señalada, se cargan todos los productos QJUMP. A parte de tener que cargar "wman" después de "ptr_gen", el orden de los ficheros no tiene importancia. Como siempre, la llamada HOTKEY debe venir después. La línea 200 y las siguientes son necesarias para el correcto funcionamiento del Toolkit del Puntero (QPTR).

5/ QRAM y el Editor de Digital Precision

```
100 base=RESPR(6074):LBYTES flp1_xtras,base:CALL base
110 base=RESPR(12388):LBYTES flp1_ptr_gen,base:CALL base
120 base=RESPR(7762):LBYTES flp1_wman,base:CALL base
130 base=RESPR(29538):LBYTES flp1_hotkey,base:CALL base
140 HOTKEY
```

El Editor de Digital Precision está escrito en SuperBasic y necesita el Turbo Toolkit para funcionar, de manera que en el fichero BOOT se carga el fichero "xtras". Se puede comenzar una copia del editor en cualquier momento ejecutándola desde el SuperBasic mediante la sentencia:

EXEC flp1_edt_bin

Por supuesto, el editor puede también ser ejecutado a partir del menú "Ficheros" de QRAM.

6/ QRAM y el sistema "runtime" de Q_Liberator y sus extensiones

```
100 base=RESPR(10016):LBYTES flp1_qlib_run,base:CALL base
110 base=RESPR(1928):LBYTES flp1_qlib_bin,base:CALL base
120 base=RESPR(1476):LBYTES flp1_qlib_ext,base:CALL base
140 base=RESPR(12388):LBYTES flp1_ptr_gen,base:CALL base
150 base=RESPR(7762):LBYTES flp1_wman,base:CALL base
160 base=RESPR(29538):LBYTES flp1_hotkey,base:CALL base
170 HOTKEY
```

Este ejemplo carga las extensiones usadas para ejecutar el compilador Q_Liberator, el cual puede entonces ser ejecutado de la forma detallada en su manual. Como se carga también el sistema "runtime", cualquier programa que no lo incluya puede ser también ejecutado.

Con QRAM se incluye una utilidad llamade BOOT_MAKE que puede usarse para acelerar la carga de las extensiones residentes colocándolas todas juntas en único fichero, el cual se cargará más rapidamente que los ficheros individuales por separado. Puede haber, como efecto secundario, una pequeña reducción de la cantidad de memoria requerida.

BOOT_MAKE produce dos ficheros: un fichero en SuperBasic nombrado normalmente "flp1_boot", y el fichero con las extensiones residentes que recibirá el mismo nombre pero acabado en "_rext". Los ficheros de extensiones pueden ser copiados desde cualquier número de medios fuente al fichero "_rext", cambiando el medio fuente si fuera necesario; como los ficheros son siempre escritos en el fichero "_rext" del medio de destino, éste debe permanecer en el mismo dispositivo hasta que BOOT_MAKE acabe. El diálogo para producir un fichero

BOOT equivalente al descrito en el anterior ejemplo 5 podría ser el siguiente:

```
Boot filename> flp1_boot
Command (ESC to finish)>
Extension file (ESC to finish)> flp2_xtras
Extension file (ESC to finish)> flp2_ptr_gen
Extension file (ESC to finish)> flp2_wman
Extension file (ESC to finish)> flp2_hotkey
Extension file (ESC to finish)>
Command (ESC to finish)> hotkey
Command (ESC to finish)>

Nombre del fichero Boot> flp1_boot
Comando (ESC para acabar)>
Fichero de extensiones (ESC para acabar)> flp2_xtras
Fichero de extensiones (ESC para acabar)> flp2_ptr_gen
Fichero de extensiones (ESC para acabar)> flp2_wman
Fichero de extensiones (ESC para acabar)> flp2_hotkey
Fichero de extensiones (ESC para acabar)>
Comando (ESC para acabar)> hotkey
Comando (ESC para acabar)>
```

El fichero BOOT resultante sería:

```
100 base=RESPR(52106):LBYTES flp1_boot_rext,base:CALL base
110 HOTKEY
```

QJUMP Ltd, 24 King Street, Ramton, Cambs.CB4 4Q (Reino Unido)

MANUAL TRADUCIDO DEL INGLES POR E. DE JESUS

Club Español de Usuarios del QL
Apartado de Correos 403
50080-ZARAGOZA

FORTH-83
EXTRA STACK

El verdadero autor de todas estas rutinas es Victor H. Yngve - Chicago, Illinois. Han sido publicadas en el Forth Dimensions Volume X, Number 3 September/October 1988. Hay permiso de FIG de reproducción para fines no comerciales (QLave es una asociación Cultural).

Hay muchos usos para el uso de un extra stack. Puede ser usado como un almacenamiento temporal en lugar del retorno stack en casos donde >R y R> no pueden ser usados.
La constante XSIZE no tiene que ser siempre 3, sino simplemente el espacio que deseemos reservar para el nuevo Stack.

Creo que con solamente ver los comentarios, el lector se deberia enterar para que sirve cada palabra definida a continuación (aunque si hay dudas, siempre le puede quedar el recurso de escribir a la sección preguntas).

```
3 CONSTANT XSIZE                ( Especifica el número de palabras en el stack )

CREATE XSTACK                               ( Define stack array )
       HERE , XSIZE 2 * ALLOT

: XCLEAR   ( -- )                ( Limpia XSTACK )
        XSTACK DUP ! ;
```

```
: >X  ( val -- )                   ( Introduce val dentro XSTACK )
      2 XSTACK +! XSTACK @ ! ;

: X> ( -- val )                    ( Saca val de XSTACK )
     XSTACK @ @ -2 XSTACK +! ;

: X@ ( -- val )                    ( pone en TOS  val de XSTACK )
      XSTACK @ @ ;

: XPICK ( n -- val )     ( pone en TOS el valor que ocupa el lugar n de XSTACK )
         XSTACK @ SWAP 2 * - @ ;

: XDEPTH ( --  n ) (  Deja el número de palabras almacenadas en XSTACK  en TOS )
       XSTACK @ XSTACK - 2/ ;
```

Las siguientes definiciones son las mismas que antes, pero chequean los limites del XSTACK por si al usuario se le ocurre hacer burradas (algo muy común en la programación).

```
( Extra stack con opcional debugging )

: X? ( pequeño grande -- ) U< NOT ABORT" XSTACK LIMITE " ;

: >X XSTACK @ XSTACK XSIZE 2 * + X?    ( Chequea max addr )
     2 XSTACK +! XSTACK @ ! ;

: X> XSTACK @ XSTACK OVER X?    ( Chequea contra min addr )
     @ -2 XSTACK +! ;

: X@ XSTACK @ XSTACK OVER X?     ( Chequea contra min addr )
      @ ;

: XPICK XSTACK @ SWAP 2 * -
       DUP XSTACK @ 2+ X?          ( Chequea contra stack top )
       XSTACK OVER X? @ ;          ( Chequea contra min addr )

: .X CR ." XSTACK: "
     XSTACK DUP @ =
     IF   ." Vacio "
     ELSE  XSTACK @ XSTACK DO I 2+ @ . 2 +LOOP
     THEN CR ;
```

A partir de aqui las rutinas son mias. Es muy posible que necesitemos guardar palabras largas o 48 bytes (los famosos números en coma flotante), pero no es muy dificil.

```
: >2X  >X >X ;         ( Introduce una palabra larga dentro de XSTACK )

: 2X>  X> X> ;         ( Saca una palabra larga de XSTACK )

: >FPX  >X >X >X ;     ( Introduce un  FP dentro de XSTACK )

: FPX>  X> X> X> ;     ( Saca un FP de XSTACK )
```

Como más de algún observador ha podido comprobar, se ha usado similares procedimientos para crear el stack extra que los usados para crear matrices en uno de mis articulos anteriores, pero la potencia del Forth está ahi, podemos hacerlo todo muy fácilmente.

Salvador Merino
Fuengirola (QLave-154).

File: cuq2_doc
Translated by Quill-View 0.5 Beta (compiled Oct 13 2008)